Packed Homomorphic Encryption Based on Ideal Lattices and Its Application to Biometrics

Masaya Yasuda[1], Takeshi Shimoyama[1], Jun Kogure[1], Kazuhiro Yokoyama[2], and Takeshi Koshiba[3]

[1] FUJITSU LABORATORIES LTD.,
1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki, 211-8588, Japan
{yasuda.masaya,shimo-shimo,kogure}@jp.fujitsu.com
[2] Department of Mathematics, Rikkyo University,
Nishi-Ikebukuro, Tokyo 171-8501, Japan
kazuhiro@rikkyo.ac.jp
[3] Division of Mathematics, Electronics and Informatics,
Graduate School of Science and Engineering, Saitama University,
255 Shimo-Okubo, Sakura, Saitama, 338-8570, Japan
koshiba@mail.saitama-u.ac.jp

Abstract. Among many approaches for privacy-preserving biometric authentication, we focus on the approach with homomorphic encryption, which is public key encryption supporting some operations on encrypted data. In biometric authentication, the Hamming distance is often used as a metric to compare two biometric feature vectors. In this paper, we propose an efficient method to compute the Hamming distance on encrypted data using the homomorphic encryption based on ideal lattices. In our implementation of secure Hamming distance of 2048-bit binary vectors with a lattice of 4096 dimension, encryption of a vector, secure Hamming distance, and decryption respectively take about 19.89, 18.10, and 9.08 milliseconds (ms) on an Intel Xeon X3480 at 3.07 GHz. We also propose a privacy-preserving biometric authentication protocol using our method, and compare it with related protocols. Our protocol has faster performance and shorter ciphertext size than the state-of-the-art prior work using homomorphic encryption.

Keywords: somewhat homomorphic encryption, ideal lattices, packed ciphertexts, secure Hamming distance, privacy-preserving biometrics.

1 Introduction

Biometric authentication (or biometrics) is an identification of clients by their physical characteristics such as fingerprint, iris, vein and DNA. Since biometric authentication has the advantage that clients do not need to remember their long and complex passwords compared to the commonly used ID/password authentication, the use of biometric authentication is now expanding (see US-VISIT [29] for a typical example). On the other hand, concerns about the security and the privacy are increasing. Especially, it is important to protect *templates*, which

A. Cuzzocrea et al. (Eds.): CD-ARES 2013 Workshops, LNCS 8128, pp. 55–74, 2013.

are stored biometric feature data, since once leaked templates can be neither revoked nor replaced. At present, there are the following three main approaches for privacy-preserving biometric authentication (see [1] or [17] for the details):

- *Feature transformation approach*, in which biometric feature data are transformed to random data by using a client-specific key or password. Cancelable biometrics [26] and biohashing [1, Section 3.3] are typical examples in this approach. This approach is practical in performance, but it is no longer secure if the client-specific key is compromised.
- *Biometric cryptosystem approach*, which is based on error-correcting codes. This approach includes fuzzy vault [19] and fuzzy commitment [20]. Since this approach needs to have strong restriction of authentication accuracy, both practical and security issues are controversial.
- *Homomorphic encryption approach*, on which we focus in this paper. In this approach, biometric feature data are protected by homomorphic encryption, and similarity of two feature data is measured on encrypted data by metrics such as the Hamming and the Euclidean distances. This approach enables biometric authentication system to be considerably secure as long as the secret key is securely managed by the trusted party. The performance and the encrypted data size are main issues for the practical use of this approach.

1.1 Related Work on Homomorphic Encryption Approach

We summarize privacy-preserving biometric authentication protocols known so far based on homomorphic encryption approach. In 2006, Schoenmakers and Tuyls in [27] proposed secure computations suitable for privacy-preserving biometric authentication using the Paillier scheme [24], which is additively homomorphic. In 2010, Osadchy et al. in [23] designed a new face recognition algorithm and proposed an efficient secure face identification system, called *SCiFI*, with the Paillier scheme and the oblivious transfer protocol. Their secure two-party computation is based on the work in [18]. In SCiFI, a feature vector extracted from face image is always represented as a binary vector of 900-bit, and the Hamming distance is used as a metric to compare two feature vectors. Their implementation showed that it took 310 ms to compute their secure Hamming distance. At present, SCiFI is known as one of the state-of-the-art privacy-preserving biometric authentication systems suitable for real life. In 2011, Blanton and Gasti in [2] developed secure protocols for iris and fingerprints. Their secure computation is similar to SCiFI, but they use the DGK scheme [10], which is an additively homomorphic encryption with shorter ciphertexts than the Paillier scheme. In their protocol, an iris feature vector is always represented as a binary vector of 2048-bit and the Hamming distance is used as in SCiFI. Their implementation showed that it took 150 ms to compute their secure Hamming distance.

1.2 Our Contributions

After Gentry's breakthrough work [13] of constructing a fully homomorphic encryption (FHE) scheme, three main variants of FHE have been proposed so far; one

based on ideal lattices [13, 14], another one based on integers [9, 11], and the last one based on the ring learning with errors (ring-LWE) assumption [5–7]. Those FHE schemes start from a somewhat homomorphic encryption (SHE) scheme, which can support only limited number of additions and multiplications on encrypted data but can be much more practical than FHE. To achieve faster secure Hamming distance, we rather use the SHE scheme based on ideal lattices (it is faster and easier to implement than the other SHE schemes). We propose an implementation of Gentry's scheme [13] for applying it to biometrics. Our variant is based mainly on Gentry-Halevi's [14], but is somewhat different from it because ours is tailored to faster secure computation. We also note that our variant is still provably secure in the sense of IND-CPA under the assumption that the ideal coset problem (ICP) [13, Section 3.2] is intractable. In this work, we will not refer to how to generate feature vectors as in [27], and assume that feature vectors are always represented as binary vectors of 2048-bit, whose length can be applied to various biometrics. Our main contributions are as follows:

- **Packing method for secure Hamming distance.** When we encrypt a feature vector bit by bit, we need to handle a large number of ciphertexts only for one feature vector and hence it would take much time to compute the Hamming distance on encrypted data. In contrast, we propose a new method to pack a feature vector into a single ciphertext, which also enables us to compute secure Hamming distance efficiently (our packing method can be applied in the SHE scheme based on the ring-LWE assumption, and results using the ring-LWE based scheme will be discussed in our next paper).
- **Privacy-preserving protocol using the SHE scheme.** We propose a new privacy-preserving biometric authentication protocol using our variant SHE scheme. We also give concrete parameters of our variant scheme with reasonable security, and demonstrate the efficiency of our packing method. Our implementation result shows that our protocol has both faster performance and shorter size of encrypted feature vectors than the state-of-the-art prior work. Furthermore, we believe that our protocol could give a new template protection technique due to our asymmetric packing methods to encrypt a feature vector in enrollment and authentication phases.

Comparison with Known Packing Methods. Smart and Vercauteren in [28] propose a packing method based on polynomial-CRT (Chinese Remainder Theorem) for packing many elements in a single ciphertext, which can be applied to perform SIMD (Single Instruction - Multiple Data) operations on encrypted data. The polynomial-CRT packing method is applied in the work [15] to evaluate the AES circuit homomorphically with a leveled FHE scheme of [5]. Furthermore, while the polynomial-CRT packing method can be applied only in ring-LWE based schemes, Brakerski, Gentry and Halevi extend the SIMD notions to the standard LWE based scheme of [7] using the packing method of [25]. Unlike the polynomial-CRT packing method, our method cannot be applied for SIMD operations, but it is very easier to handle and much more efficient for evaluating fundamental computations such as the Hamming distance (it would

be more interesting to combine our packing method with the polynomial-CRT method). In their work [21], Lauter, Naehrig and Vaikuntanathan also present some message encoding techniques in the ring-LWE based scheme, and their technique is to encode integers in a single ciphertext so that it enables us to efficiently compute their sums and products over the integers. When we ignore the difference of homomorphic encryption schemes, our packing method can be considered as an extension of their techniques. Our extension is to give two types of packed ciphertexts, and combinations of two types of our packed ciphertexts give efficient computations such as the inner product and the Hamming distance.

2 Preliminaries

We fix our standard notation. The symbols $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ denote the ring of integers, the field of rational numbers, the field of real numbers, and the field of complex numbers, respectively. For a prime number p, the finite field with p elements is denoted by $\mathbb{F}_p$. For two integers z and d, let $[z]_d$ denote the reduction of z modulo d included in the interval $[-d/2, d/2)$ as in [14] (let $z \bmod d$ denote the usual reduction included in the interval $[0, d)$). For a rational number $q \in \mathbb{Q}$, we denote by $\lceil q \rfloor$ the rounding of q to the nearest integer, and by $[q]$ the fractional part of q. These notations are extended to vectors and matrices in the natural way. For a vector $\boldsymbol{a} = (a_1, a_2, \ldots, a_n) \in \mathbb{R}^n$, let $||\boldsymbol{a}||$ denote the Euclidean norm defined by $\sqrt{\sum_{i=1}^n a_i^2}$. Furthermore, we let $||\boldsymbol{a}||_1$ and $||\boldsymbol{a}||_\infty$ denote the 1-norm defined by $\sum_{i=1}^n |a_i|$ and ∞-norm defined by $\max_i |a_i|$, respectively.

2.1 Definitions and Notation in Lattices

Fix an integer number n. Let $B \in \mathbb{R}^{n \times n}$ be a matrix and let $\mathbf{b}_i \in \mathbb{R}^n$ denote the i-th row of B for $i = 1, \ldots, n$. Denote by

$$\mathcal{L}(B) = \left\{ \sum_{i=1}^{n} m_i \mathbf{b}_i : m_i \in \mathbb{Z} \right\}$$

the set of all integral linear combinations of the $\mathbf{b}_i$'s, which is a subgroup of $\mathbb{R}^n$. We say that the subgroup $\mathcal{L}(B)$ is a (full-rank) lattice of dimension n if $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are linearly independent. In this case, we say that the matrix B is a basis of the lattice $\mathcal{L}(B)$. Every lattice has infinitely many lattice bases. If B_1 and B_2 are two bases of a lattice L, then there exists a unimodular matrix $U \in \mathrm{GL}_n(\mathbb{Z})$ satisfying $B_1 = UB_2$. Since we have $\det(U) = \pm 1$, the absolute value $|\det(B)|$ is invariant for any basis B of L and denoted by $\det(L)$. For a basis B, we let

$$\mathcal{P}(B) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in [-1/2, 1/2) \right\}$$

denote its associated half-open parallelepiped. Every lattice L has a unique Hermite normal form basis $\mathrm{HNF}(L) = (b_{ij})$, where $b_{ij} = 0$ for all $i < j$, $b_{jj} > 0$ for

all j, and $b_{ij} \in [-b_{jj}/2, b_{jj}/2)$ for all $i > j$. Given any basis of L, we can compute the basis $\mathrm{HNF}(L)$ by Gaussian elimination. Note that the basis $\mathrm{HNF}(L)$ typically serves as the public key representation of the lattice.

By lattice reduction, we mean an operation that computes a basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]^t$ of L with short and nearly orthogonal vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ from a given basis of L. Lattice reduction algorithms are often used for breaking lattice cryptosystems. The *root Hermite factor* of a lattice reduction algorithm is defined by $||\mathbf{b}_1||/\det(L)^{1/n}$ with the output basis $[\mathbf{b}_1, \ldots, \mathbf{b}_n]^t$. It is an index to measure the output quality of a lattice reduction algorithm (the output quality is better as the root Hermite factor is smaller). The most practical lattice reduction algorithms are the LLL and the BKZ algorithms.

2.2 Basic Construction of SHE Scheme

We present the basic construction of our variant of the SHE scheme based on ideal lattices (see §3 for some improvements). Our variant is based mainly on Gentry-Halevi's [14], but ours can use a more general ciphertext space for faster secure computation. For a 2-power integer $n = 2^m$, let $R := \mathbb{Z}[x]/(f_n(x))$ denote the polynomial ring modulo $f_n(x) := x^n + 1$, which is an irreducible polynomial. Since a map

$$\ni v(x) = v_0 + v_1 x + \cdots v_{n-1} x^{n-1} \mapsto \boldsymbol{v} = (v_0, v_1, \ldots, v_{n-1}) \in \mathbb{Z}^n \qquad (1)$$

gives an isomorphism $R \simeq \mathbb{Z}^n$ as $\mathbb{Z}$-modules, we can view each element of R as both a polynomial $v(x)$ and a vector $\boldsymbol{v}$.

Key Generation. To generate the public and the secret keys, we need key parameters (n, t, s), where $n = 2^m$ is the lattice dimension of 2-power, t is the bit length of coefficients in so called the generating polynomial $v(x)$, and s is the size of the plaintext space. The following construction is based on the sub-optimal key generation described in [14, Section 3] (see §3.3 for our improved key generation):

Step 1. We first choose an n-dimensional vector $\boldsymbol{v} = (v_0, v_1, \ldots, v_{n-1}) \in \mathbb{Z}^n$, where v_i is randomly chosen satisfying the condition $|v_i| \leq 2^t$ for any i. Set $v(x) = \sum_{i=0}^{n-1} v_i x^i \in R$ as a generating polynomial. Consider the rotation matrix

$$V := \mathrm{rot}(\boldsymbol{v}) = \begin{pmatrix} v_0 & v_1 & v_2 & \cdots & v_{n-1} \\ -v_{n-1} & v_0 & v_1 & \cdots & v_{n-2} \\ -v_{n-2} & -v_{n-1} & v_0 & \cdots & v_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -v_1 & -v_2 & -v_3 & \cdots & v_0 \end{pmatrix}. \qquad (2)$$

Since the i-th row of V corresponds to the polynomial $v(x) \times x^i \in R$ under the isomorphism (1), the subgroup $L := \mathcal{L}(V) \subset \mathbb{Z}^n$ is a lattice of dimension n and we have the relation $R \supset (v(x)) \simeq L \subset \mathbb{Z}^n$, where $(v(x))$ denotes the principal ideal of R generated by $v(x)$.

Step 2. By applying the extended Euclidean-GCD algorithm for polynomials, we compute the scaled inverse $w(x)$ of $v(x)$ modulo $f_n(x)$ satisfying

$$w(x) \times v(x) \equiv d \bmod f_n(x).$$

Note that d is the resultant of $v(x)$ and $f_n(x)$, which is also equal to the determinant $\det(L) = |\det(V)|$ of the lattice L. If $\gcd(d, s) \neq 1$, go back to Step 1 and generate another $\boldsymbol{v}$ (we can decrypt a ciphertext without the secret key when s divides d). Let $\boldsymbol{w} = (w_0, w_1, \ldots, w_{n-1})$ denote the vector corresponding to $w(x)$. Then the matrix $W := \mathrm{rot}(\boldsymbol{w})$ satisfies $W \times V = V \times W = d \cdot I$, where I is the $n \times n$ identity matrix.

Step 3. We give the following definition and lemma given in [14, Section 3]:

Definition 1 (goodness of $v(x)$). *We say that $v(x)$ is good if the Hermite normal form basis $B := \mathrm{HNF}(L)$ of the lattice $L = \mathcal{L}(V)$ has the form*

$$B = \begin{pmatrix} d & 0 & 0 & \cdots & 0 \\ -r & 1 & 0 & \cdots & 0 \\ * & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ * & 0 & 0 & \cdots & 1 \end{pmatrix}. \tag{3}$$

Lemma 1. *A generating polynomial $v(x)$ is good if and only if L contains a vector of the form $(-r', 1, 0, \ldots, 0)$. Furthermore, if $v(x)$ is good, we have that $r := w_1/w_0 = w_2/w_1 = \cdots = w_{n-1}/w_{n-2} = -w_0/w_{n-1} \bmod d$ and the element r satisfies the condition $r^n \equiv -1 \bmod d$.*

In this step, we check whether $v(x)$ is good or not. For checking it, we only test that $r := w_1/w_0 \bmod d$ satisfies $r^n \equiv -1 \bmod d$; If $r^n \equiv -1 \bmod d$, go to the next step. Otherwise, go back to Step 1 and generate another $\boldsymbol{v}$.

Step 4. We set V, W (resp. B) as the secret key (resp. the public key). We here call V, W (resp. B) the *secret key matrices* (resp. the *public key matrix*). Due to the special form (3) of B, we only need to set $\mathsf{sk} = w_i$ as the secret key and $\mathsf{pk} = (d, r, n, s)$ as the public key, where w_i is a single coefficient of $\boldsymbol{w}$ satisfying $\gcd(w_i, s) = 1$ (see Decryption below).

Encryption. To encrypt a plaintext $b \in \mathbb{Z}/s\mathbb{Z} = \{0, 1, \ldots, s-1\}$ with $\mathsf{pk} = (d, r, n, s)$, we first choose a random "noise vector" $\boldsymbol{u} = (u_0, u_1, \ldots, u_{n-1})$ with $u_i \in \{0, \pm 1\}$ chosen as 0 with some probability q and as ± 1 with probability $(1-q)/2$ each. Then the ciphertext of b is given by the integer

$$\mathsf{Enc}(b) = \left[b + s \sum_{i=0}^{n-1} u_i r^i \right]_d.$$

Set $\boldsymbol{a} := s\boldsymbol{u} + b\boldsymbol{e}_1 = (su_0 + b, su_1, \ldots, su_{n-1})$ with $\boldsymbol{e}_1 = (1, 0, \ldots, 0)$, and let $a(x) \in R$ denote the corresponding polynomial. Then we have $\mathsf{Enc}(b) = [a(r)]_d$ and the vector $(\mathsf{Enc}(b), 0, \ldots, 0)$ is equal to $\boldsymbol{a} \bmod B := \boldsymbol{a} - \left(\lceil \boldsymbol{a} \times B^{-1} \rfloor \times B \right) \in \mathcal{P}(B)$, which is the ciphertext vector generated by the public key matrix B (see [14, Section 5] for details).

Definition 2 (masked plaintext). *We call the vector* $\boldsymbol{a}$ *(or the polynomial* $a(x)$*) the masked plaintext corresponding to a ciphertext* ct.

We need to choose the probability q to make it hard to recover the original noise vector from a ciphertext c. Against exhaustive-search and birthday attacks, we set a security parameter λ, where we need to set q satisfying $2^{(1-q)n} \cdot \binom{n}{qn} > 2^{2\lambda}$ [14, Section 5.2]. Furthermore, Gentry and Halevi in [14] considered the hybrid attack, whose method is to choose a random subset of the powers of r including all the noise coefficients and search for a small vector in this low-dimension lattice (e.g., dimension 200). It is sufficient to set q satisfying $\left(\frac{n}{200}\right)^{qn} \geq 2^{\lambda}$ against the hybrid attack. For $\lambda = 80$ and $n \geq 1024$, the above two inequalities are satisfied if $q = \frac{1}{3}$. In this paper, we fix

$$q = \frac{1}{3}$$

for higher security. In contrast, Gentry and Halevi in [14] take an aggressive setting where the number of nonzero entries in the noise vectors is between 15 and 20 for FHE public challenges).

Decryption. To decrypt a ciphertext $\mathsf{Enc}(b)$ with the secret key matrices V, W, we first recover the corresponding masked plaintext by $\boldsymbol{a} = \boldsymbol{c} \bmod V = \boldsymbol{c} - \left(\lceil \boldsymbol{c} \times V^{-1} \rfloor \times V\right) = [\boldsymbol{c} \times W/d] \times V$ with the ciphertext vector $\boldsymbol{c} = (\mathsf{Enc}(b), 0, \ldots, 0)$. It follows from [14, Section 6] that we can recover the masked plaintext $\boldsymbol{a}$ if every entry in $\boldsymbol{a} \times W$ is less than $d/2$ in absolute value. For $\boldsymbol{a} = (a_0, a_1, \ldots, a_{n-1})$, we then output $b = a_0 \bmod s \in \mathbb{Z}/s\mathbb{Z}$ as the decryption result.

In [14, Section 6.1], Gentry and Halevi proposed an optimized decryption procedure in the case $s = 2$. We can extend their method to our variant scheme; Let $\boldsymbol{a} = s\boldsymbol{u} + b\boldsymbol{e}_1$ be the masked plaintext of a ciphertext $\mathsf{Enc}(b)$. From a similar argument of [14, Section 6.1], we have

$$[\boldsymbol{c} \times W]_d = \boldsymbol{a} \times W = s\boldsymbol{u} \times W + b \cdot (w_0, w_1, \ldots, w_{n-1})$$

if every entry in $\boldsymbol{a} \times W$ is less than $d/2$ in absolute value. Since $[\boldsymbol{c} \times W]_d = ([\mathsf{Enc}(b) \cdot w_0]_d, [\mathsf{Enc}(b) \cdot w_1]_d, \ldots, [\mathsf{Enc}(b) \cdot w_{n-1}]_d)$, we have $[\mathsf{Enc}(b) \cdot w_i]_d \equiv b \cdot w_i \bmod s$ for any i. It is therefore sufficient to keep one coefficient w_i of $\boldsymbol{w}$ with $\gcd(w_i, s) = 1$ as the secret key sk, and then we can recover b by computing

$$[\mathsf{Enc}(b) \cdot \mathsf{sk}]_d \cdot \mathsf{sk}^{-1} \bmod s. \tag{4}$$

Note that there always exists w_i satisfying $\gcd(w_i, s) = 1$ if we take $s = 2^k$.

Homomorphic Operations. For two ciphertexts $\mathsf{Enc}(b_1)$ and $\mathsf{Enc}(b_2)$, the homomorphic addition "$\dot{+}$" is defined by

$$\mathsf{Enc}(b_1) \dot{+} \mathsf{Enc}(b_2) := [\mathsf{Enc}(b_1) + \mathsf{Enc}(b_2)]_d.$$

The homomorphic subtraction is also defined by $\mathsf{Enc}(b_1) - \mathsf{Enc}(b_2) := [\mathsf{Enc}(b_1) - \mathsf{Enc}(b_2)]_d$. Similarly, the homomorphic multiplication "$*$" is defined by

$$\mathsf{Enc}(b_1) * \mathsf{Enc}(b_2) := [\mathsf{Enc}(b_1) \cdot \mathsf{Enc}(b_2)]_d.$$

Let $\boldsymbol{a}_1, \boldsymbol{a}_2 \in R$ denote the masked plaintexts corresponding to $\mathsf{Enc}(b_1), \mathsf{Enc}(b_2)$, respectively. Then we see that the vectors $([\mathsf{Enc}(b_1) + \mathsf{Enc}(b_2)]_d, 0, \ldots, 0)$ and $([\mathsf{Enc}(b_1) \cdot \mathsf{Enc}(b_2)]_d, 0, \ldots, 0)$ are equal to $\boldsymbol{a}_1 + \boldsymbol{a}_2 \bmod B$ and $\boldsymbol{a}_1 \times \boldsymbol{a}_2 \bmod B$ respectively, where "$\times$" denotes the multiplication operation in the ring R. This shows that the homomorphic operations correspond to the ring structure of R, from which the homomorphic property of our variant scheme follows. However, homomorphic operations make the size of the noise vector in the corresponding masked plaintext larger. Therefore it is only possible to add and multiply ciphertexts before the size of the noise vector grows beyond the decryption range.

3 Some Improvements of SHE Scheme

In this section, we give some improvements of our variant SHE scheme described in §2.2, mainly for an efficient computation of secure Hamming distance.

3.1 Theoretical Evaluation of Decryption Range

In applying the SHE scheme to a concrete application scenario, the size of its decryption range is the most important in choosing key parameters suitable for the scenario. Under the condition $|v_i| \leq 2^t$, Gentry and Halevi in [14, Section 7] experimentally estimate that the decryption range is roughly equal to 2^t and it succeeds to decrypt a ciphertext ct if the corresponding masked plaintext $\boldsymbol{a}$ satisfies $||\boldsymbol{a}|| \leq 2^t$. In contrast, we give a theoretical evaluation of the range under a certain condition in choosing a generating polynomial $v(x) = \sum_{i=0}^{n-1} v_i x^i \in R$ (cf. Key Generation in §2.2). In the following, we give our technical result without a proof due to lack of space (a complete proof will be given in our forthcoming paper):

Proposition 1. *Assume the condition*

$$(\clubsuit): \quad T = |v_{n-1}| = 2^t(1 + \varepsilon_{n-1}) \text{ and } v_i = T\varepsilon_i \text{ with } |\varepsilon_i| < \tfrac{1}{4n} \text{ for } 0 \leq i \leq n-2 \text{ and } \varepsilon_{n-1} > 0.$$

Then the decryption of a ciphertext ct *succeeds if the corresponding masked plaintext* $\boldsymbol{a}$ *satisfies either*

$$||\boldsymbol{a}||_1 < \frac{11 \cdot 2^{t-1}}{13} \quad \text{or} \quad ||\boldsymbol{a}||_\infty < \frac{11n \cdot 2^{t-1}}{19n - 6}. \tag{5}$$

Since we have $|v_{n-1}| = T \approx 2^t$ and $|v_i| < T/4n$ for $0 \leq i \leq n-2$ under the condition $(\clubsuit)$ for sufficiently small $\varepsilon_{n-1} > 0$, we can consider our condition as a restricted version of Gentry-Halevi's condition $|v_i| \leq 2^t$. Our theoretical evaluation gives the same level as Gentry-Halevi's experimental one, and especially it enables us to use the ∞-norm evaluation, which is independent of the noise probability q in encryption (Gentry-Halevi's evaluation uses the Euclidean norm, and it is deeply related with q). Hence our evaluation makes it easier to choose key parameters in applying our variant SHE scheme to a concrete scenario (in particular, the ∞-norm evaluation makes it easy in the noise management of ciphertexts).

3.2 Packing Method for Secure Hamming Distance

As described in Encryption of §2.2, we can only encrypt an element of $\mathbb{Z}/s\mathbb{Z}$ in our variant scheme. In contrast, we propose a new method to pack a binary vector in a single ciphertext, which is also suitable for an efficient computation of both secure inner product and secure Hamming distance. Our packing method is to transform a binary vector into a certain polynomial of the ring R, and to encrypt the transformed polynomial. Furthermore, for efficient secure computation, we make use of polynomial operations in the ring R. As described in §1.2, our packing method can be considered as an extension of the method proposed in [21], and our main extension is to give two types of packed ciphertexts. However, our packing method causes a difficulty that it is not straightforward to apply the optimized decryption procedure (4) described in Decryption of §2.2. To overcome the problem, we need to take a generating polynomial $v(x)$ satisfying a certain condition (see Proposition 3 below). We begin with our packing method and efficient computation of the secure inner product. As remarked in §1.2, we only consider 2048-bit binary vectors for securing the Hamming distance often used in biometrics.

Definition 3 (packing method vEnc_i). *Assume $n \geq 2048$. We define two types of packing method as follows (cf. Encryption of §2.2):*

Type 1. *For a binary vector A, let $F_1 : A = (A_0, \ldots, A_{2047}) \mapsto \sum_{i=0}^{2047} A_i x^i \in R = \mathbb{Z}[x]/(f_n(x))$. Then its packed ciphertext is defined by the integer*

$$\mathsf{vEnc}_1(A) := [F_1(A)(r) + su_1(r)]_d = \left[\sum_{i=0}^{2047} A_i r^i + su_1(r)\right]_d \tag{6}$$

using the public key $\mathsf{pk} = (d, r, n, s)$, where $u_1(x) \in R$ denotes a noise polynomial. We note that the masked plaintext of $\mathsf{vEnc}_1(A)$ corresponds to the polynomial $F_1(A) + su_1(x) \in R$.

Type 2. *For a binary vector B, let $F_2 : B = (B_0, \ldots, B_{2047}) \mapsto -\sum_{i=0}^{2047} B_i x^{n-i}$. Then its packed ciphertext is defined by the integer*

$$\mathsf{vEnc}_2(B) := [F_2(B)(r) + su_2(r)]_d = \left[-\sum_{i=0}^{2047} B_i r^{n-i} + su_2(r)\right]_d, \tag{7}$$

where $u_2(x)$ denotes a noise polynomial. We also note that the masked plaintext of $\mathsf{vEnc}_2(B)$ corresponds to the polynomial $F_2(B) + su_2(x) \in R$.

Proposition 2 (secure inner product). *Assume $n \geq 2048$. Let ct be the ciphertext given by the homomorphic multiplication of $\mathsf{vEnc}_1(A)$ and $\mathsf{vEnc}_2(B)$. Let $\boldsymbol{a} = (a_0, \ldots, a_{n-1}) \in R = \mathbb{Z}^n$ denote the masked plaintext corresponding to ct. Then we have*

$$a_0 \equiv \sum_{i=0}^{2047} A_i B_i \bmod s.$$

Proof. The homomorphic multiplication corresponds to the multiplication in the ring R. This shows that $\boldsymbol{a} = (F_1(A) + su_1(x)) \times (F_2(B) + su_2(x)) \in R$ and we have $\boldsymbol{a} \equiv F_1(A) \times F_2(B) \bmod s$. We easily see that the constant term of

$$F_1(A) \times F_2(B) = \sum_{i=0}^{2047} A_i x^i \times \sum_{j=0}^{2047} B_j x^{n-j}$$

is equal to $\sum_{i=0}^{2047} A_i B_i$ since $x^n = -1$ in the ring R. □

Let ct and $\boldsymbol{a} = (a_0, \ldots, a_{n-1})$ be as in Proposition 2. By Proposition 2, we only need to recover $a_0 \bmod s$ with the secret key sk to obtain the inner product. However, it is not straightforward to apply the optimized decryption procedure (4) described in Decryption of §2.2 since the masked plaintext $\boldsymbol{a} \in R$ is not of the form $s\boldsymbol{u} + b\boldsymbol{e}_1$, which we handle for the usual encryption $\mathsf{Enc}(b)$ described in Encryption of §2.2. To apply the optimized decryption (4) for the recovery of $a_0 \bmod s$ with sk, we need the following result:

Proposition 3 (suitable condition for decryption). *Let* ct *and* $\boldsymbol{a}$ *be as in Proposition 2. For a generating polynomial* $v(x) = \sum_{i=0}^{n-1} v_i x^i \in R$*, we assume* $v_i \in s\mathbb{Z}$ *for* $i = 1, \ldots, n-1$ *(but we take* $v_0 \notin s\mathbb{Z}$*). Let* $\boldsymbol{w} = (w_0, w_1, \ldots, w_{n-1})$ *be the vector generated in* Key Generation *of §2.2, and we take* w_0 *as the secret key* sk*. Then we can recover* $a_0 \bmod s$ *with* sk *by computing*

$$[\mathsf{ct} \cdot \mathsf{sk}]_d \cdot v_0 \cdot d^{-1} \bmod s.$$

In particular, we can recover $a_0 \bmod s$ *with* sk *by computing* $[\mathsf{ct} \cdot \mathsf{sk}]_d \bmod s$ *if we take* v_0 *satisfying* $v_0 \in 1 + s\mathbb{Z}$*.*

Proof. For $\boldsymbol{v} = (v_0, \ldots, v_{n-1})$, let $V = \mathrm{rot}(\boldsymbol{v})$ and $W = \mathrm{rot}(\boldsymbol{w})$ be the secret key matrices. For the ciphertext vector $\boldsymbol{c} = (\mathsf{ct}, 0, \ldots, 0)$, it follows from a similar argument of Decryption of §2.2 that

$$\begin{aligned} [\boldsymbol{c} \times W]_d = \boldsymbol{a} \times W \iff \\ ([\mathsf{ct} \cdot w_0]_d, \ldots, [\mathsf{ct} \cdot w_{n-1}]_d) \cdot V = \boldsymbol{a} \times W \times V = (da_0, \ldots, da_{n-1}) \end{aligned} \tag{8}$$

if every entry in $\boldsymbol{a} \times W$ is less than $d/2$ in absolute value (note that we have $W \times V = d \cdot I$). By comparing the first entry of the vectors in the right hand side of (8), we have $[\mathsf{ct} \cdot w_0]_d \cdot v_0 \equiv da_0 \bmod s$ by the assumption $v_i \in s\mathbb{Z}$ for $i = 1, \ldots, n-1$. Hence we can recover $a_0 \bmod s$ by computing $[\mathsf{ct} \cdot w_0]_d \cdot v_0 \cdot d^{-1} \bmod s$. In particular, if $v_0 \in 1 + s\mathbb{Z}$, we have $v_0 \cdot d^{-1} \equiv 1 \bmod s$ since $d \equiv v_0^n \equiv 1 \bmod s$ by $d = \det(V)$ and $v_i \in s\mathbb{Z}$ for $i = 1, \ldots, n-1$. This completes the proof. □

Secure Hamming Distance. We apply our packing method to compute the Hamming distance on encrypted data. Assume $n, s \geq 2048$. For the guarantee of both the theoretical decryption range of Proposition 1 and the success of the decryption, we need to take a generating polynomial $v(x)$ under the condition

$$(\spadesuit) : v_0 \in 1 + s\mathbb{Z},\ v_i \in s\mathbb{Z} \text{ for } 1 \leq i \leq n-1, \text{ and the condition } (\clubsuit).$$

Take w_0 as the secret key sk as in Proposition 3. For two binary vectors A and B, the Hamming distance $d_H(A,B)$ can be computed by $\sum_{i=0}^{2047}(A_i+B_i-2A_iB_i) = \mathrm{HW}(A)+\mathrm{HW}(B)-2\sum_{i=0}^{2047}A_iB_i$, where $\mathrm{HW}(\cdot)$ denotes the Hamming weight of a binary vector. For computing the Hamming distance on encrypted data, we consider two integers

$$C_1 := \left[\sum_{i=0}^{n-1} r^i\right]_d \quad \text{and } C_2 := [-C_1+2]_d$$

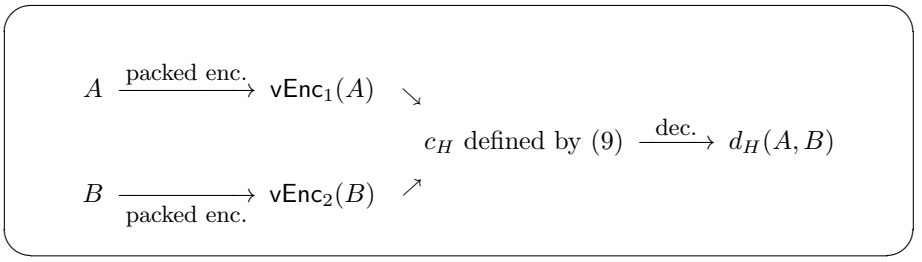
with the public key $\mathsf{pk} = (d, r, n, s)$. Set $c_1(x) := \sum_{i=0}^{n-1} x^i$ and $c_2(x) := -c_1(x)+2 = 1-\sum_{j=1}^{n-1} x^j$ in the ring R. From a similar argument in the proof of Proposition 2, the homomorphic multiplication of $\mathsf{vEnc}_1(A)$ and C_2 (resp. $\mathsf{vEnc}_2(B)$ and C_1) corresponds to the masked plaintext $(F_1(A)+su_1(x)) \times c_2(x)$ (resp. $(F_2(B)+su_2(x)) \times c_1(x)$), whose constant term modulo s is equal to $\mathrm{HW}(A)$ (resp. $\mathrm{HW}(B)$). Therefore the encrypted Hamming distance is computed by

$$\begin{aligned}\mathsf{ct}_H &= C_2 * \mathsf{vEnc}_1(A) \dotplus C_1 * \mathsf{vEnc}_2(B) \dotplus (-2\mathsf{vEnc}_1(A) * \mathsf{vEnc}_2(B)) \\ &= C_1 * (-\mathsf{vEnc}_1(A) \dotplus \mathsf{vEnc}_2(B)) \dotplus 2\mathsf{vEnc}_1(A) * (1-\mathsf{vEnc}_2(B)) \end{aligned} \tag{9}$$

which only requires two homomorphic multiplications, two homomorphic additions and one left-shift of a ciphertext if we precompute the integer C_1. For the masked plaintext $\boldsymbol{a}_H \in R = \mathbb{Z}^n$ corresponding to ct_H, we see that the first entry of $\boldsymbol{a}_H$ modulo s is equal to the Hamming distance $d_H(A,B)$ by Proposition 2. By Proposition 3, we can recover $d_H(A,B)$ with sk by computing

$$[\mathsf{ct}_H \cdot \mathsf{sk}]_d \bmod s \tag{10}$$

if $\boldsymbol{a}_H$ is included in the theoretical decryption range of Proposition 1. The following diagram represents the computation procedure of secure Hamming distance with our packing method:



Remark 1. As mentioned in §1.2, our variant is just an instantiation of Gentry's abstract scheme described in [13, Section 3.1] (of course, ours provides an "efficient" implementation). Since Gentry's scheme is provably secure in the sense of IND-CPA under the ICP assumption even in the abstract level (see [13, Section 3.2] for the detail). Thus ours inherits the provable security from Gentry's (we note that Gentry's scheme is not IND-CCA1 [22]).

3.3 Improving Key Generation

The problem of the construction of our variant scheme described in §2.2 is the slow key generation. Based on Gentry-Halevi's optimized key generation [14, Section 4], we give some improvements for faster key generation than their one, and also guarantee the theoretical decryption range of Proposition 1. The main idea of Gentry-Halevi's optimized key generation is as follows: For a lattice dimension $n = 2^m$, fix a primitive $2n$-th root ρ of $f_n(x) = x^n + 1$ and set $\rho_i = \rho^{2i+1}$ for $i = 0, 1, \ldots, n-1$, which are the roots of $f_n(x)$ in the field $\mathbb{C}$ of complex numbers. To generate the key pair $(\mathsf{pk}, \mathsf{sk})$, we consider two polynomials

$$g(z) := \prod_{i=0}^{n-1} (v(\rho_i) - z) \text{ and } h(z) := \prod_{i=0}^{n-1} (v(\rho_i) - z/\rho_i)$$

for a generating polynomial $v(x)$. Although $g(z)$ and $h(z)$ are defined by $\rho_i \in \mathbb{C}$, the coefficients of $g(z)$ and $h(z)$ are all integers. Let g_i (resp. h_i) denote the coefficient of z^i in $g(z)$ (resp. $h(z)$). By [14, Section 4], we have $d = g_0$, $w_0 = -g_1/n$ and $w_1 = -h_1/n$ if $v(x)$ is square-free. Since $w_1/w_0 = \cdots = w_{n-1}/w_{n-2} = -w_0/w_{n-1} \bmod d$ in the case where $v(x)$ is good by Lemma 1, we can recover the rest of w_i from the triple (d, w_0, w_1). After recovering the triple (d, w_0, w_1), we therefore compute the ratio $r = w_1/w_0 \bmod d$ and verify that $r^n \equiv -1 \bmod d$, which is the condition for testing that $r \cdot w_i = w_{i+1} \bmod d$. In the following, we give our improvements:

3.3.1 Simultaneous Computation of $g(z), h(z) \bmod z^2$

From the above argument, we need to consider $g(z), h(z) \bmod z^2$ to obtain the triple (d, w_0, w_1). In [14, Section 4.1], Gentry and Halevi showed a method to compute $g(z), h(z) \bmod z^2$ separately. In contrast, we give a method to compute $g(z), h(z) \bmod z^2$ at the same time: Set $U_0(x) = 1$, $V_0(x) = v(x)$ and $W_0(x) = -x^{n-1}$. For $0 \le j \le m$, let

$$\begin{cases} V_j(x) \cdot V_j(-x) \bmod f_{n_j}(x) = \sum_{i=0}^{n_j-1} a_i^{(j)} x^i, \\ U_j(x) \cdot V_j(-x) + U_j(-x) \cdot V_j(x) \bmod f_{n_j}(x) = \sum_{i=0}^{n_j-1} b_i^{(j)} x^i, \\ W_j(x) \cdot V_j(-x) + W_j(-x) \cdot V_j(x) \bmod f_{n_j}(x) = \sum_{i=0}^{n_j-1} c_i^{(j)} x^i, \end{cases}$$

and $U_{j+1}(x) = \sum_{k=0}^{n_j/2-1} b_{2k}^{(j)} x^k$, $V_{j+1}(x) = \sum_{k=0}^{n_j/2-1} a_{2k}^{(j)} x^k$, $W_{j+1}(x) = \sum_{k=0}^{n_j/2-1} c_{2k}^{(j)} x^k$, where $f_{n_j}(x) := x^{n_j} + 1$ with $n_j = n/2^j$. Then we have

$$g_j(z) := \prod_{i=0}^{n_j-1} \left(V_j(\rho_i^{2^j}) - z U_j(\rho_i^{2^j}) \right) \equiv g(z) \bmod z^2, \tag{11}$$

$$h_j(z) := \prod_{i=0}^{n_j-1} \left(V_j(\rho_i^{2^j}) - z W_j(\rho_i^{2^j}) \right) \equiv g(z) \bmod z^2 \tag{12}$$

by a similar argument of [14, Section 4.1]. Since the degree of $U_m(x)$, $V_m(x)$ and $W_m(x)$ is equal to 0, we can consider these polynomials as integers U_m, V_m and W_m respectively, and we have $g_0 = h_0 = V_m$, $g_1 = U_m$ and $h_1 = W_m$ by the equations (11) and (12). Therefore we only need to compute $U_j(x), V_j(x)$ and $W_j(x)$ to obtain the triple (d, w_0, w_1).

3.3.2 Quick Test of Goodness

In the optimal key generation of [14], after computing (d, w_0, w_1), we test the condition $r^n \equiv -1 \bmod d$ to check that $v(x)$ is not good (see also Key Generation of §2.2). In the case where $r^n \not\equiv -1 \bmod d$, we need to compute the triple (d, w_0, w_1) again for another generating polynomial $v(x)$. Since the cost of computing the triple (d, w_0, w_1) is dominant in the key generation, we consider a method to check that $v(x)$ is not good without computing the triple (d, w_0, w_1) exactly. We give the following lemma:

Lemma 2. *If $v(x)$ is good, we have $d \equiv 1 \bmod 2n$.*

Proof. Assume that $v(x)$ is good, namely, the public key matrix B has the form (3). Then we see that $f_n(x) \bmod d$ is split completely. This shows that $f_n(x) \bmod p$ is split completely for the primes $p \mid d$. By the theory of cyclotomic fields, we have that $p \equiv 1 \bmod 2n$ for the primes $p \mid d$, which shows that $d \equiv 1 \bmod 2n$. □

To check that $v(x)$ is not good, we test the condition $d \equiv 1 \bmod 2n$. In testing $d \equiv 1 \bmod 2n$, we only need to compute $d \bmod 2n$. By using $V_j(x) \bmod 2n \in (\mathbb{Z}/2n\mathbb{Z})[x]$ to compute $d \bmod 2n$, we can check it in lower cost (after obtaining the triple (d, w_0, w_1), we can check the goodness of $v(x)$ exactly by testing the condition that $v(x) \cdot w(x) \equiv d \bmod f_n(x)$, where $w(x) = \sum_{i=0}^{n-1} w_i x^i$ with $r = w_1/w_0 \bmod d$ and $w_i = r^i \cdot w_0$ for $i \geq 1$).

3.3.3 Guarantee of Theoretical Decryption Range

We give a method to choose a generating polynomial $v(x)$ in order to guarantee the theoretical decryption range of Proposition 1. In the key generation, we choose a generating polynomial $v(x) = \sum_{i=0}^{n-1} v_i x^i \in R$, where $v_i \in \mathbb{Z}$ is randomly chosen with the condition (♠) for our secure Hamming distance (see Proposition 3). Then we can use the theoretical evaluation of the decryption range of Proposition 1.

Improved Key Generation Algorithm. In Algorithm 1, we give a concrete algorithm of our key generation. Using our key generation, we can generate a good generating polynomial $v(x)$ with higher probability than the key generation of [14]. Since $v(x)$ with $d \equiv 1 \bmod 2n$ satisfies the condition $r^n \equiv -1 \bmod d$ with very high probability, we hardly go back to Step 1 in Step 4 of Algorithm 1, which shows that Step 2 has the advantage of the processing performance. n particular, our experiment shows that our algorithm is 25% faster on average than the case without Step 2 for input $(n, t, s) = (4096, 37, 2048)$.

Algorithm 1. Improved key generation of our variant SHE scheme

Input: (n, t, s): key parameters (for simplicity, assume that s is a 2-power integer)
Output: The public key $\mathsf{pk} = (d, r, n, s)$ and the secret key $\mathsf{sk} = w$.

1: Generate a random vector $\boldsymbol{v} = (v_0, v_1, \ldots, v_{n-1}) \in \mathbb{Z}^n$, where $v_i \in \mathbb{Z}$ is chosen under the condition (♣) (or (♠) for secure Hamming distance), and set $v(x) = \sum_{i=0}^{n-1} v_i x^i$.
2: By computing the followings, we verify that $d = \mathrm{res}(f_n(x), v(x)) \equiv 1 \bmod 2n$:

(2-a) $S_0(x) \leftarrow v(x) \bmod 2n \in (\mathbb{Z}/2n\mathbb{Z})[x]$.
(2-b) For $j = 0$ to $m - 1$, do

i. Set $f_{n_j}(x) = x^{n_j} + 1$ with $n_j = n/2^j$. Compute $T_j(x) = S_j(x) \cdot S_j(-x) \bmod (2n, f_{n_j}(x)) = \sum_{i=0}^{n_j-1} t_i x^i$ with $t_i \in \{0, 1, \ldots, 2n-1\}$.
ii. $S_{j+1}(x) \leftarrow \sum_{k=0}^{n_j/2-1} t_{2k} x^k$.

(2-c) Then $d \equiv S_m \bmod 2n$. If $S_m \not\equiv 1 \bmod 2n$, go back to Step 1.

3: Compute the triple (d, w_0, w_1) as follows:

(3-a) $U_0(x) \leftarrow 1$, $V_0(x) \leftarrow v(x)$, $W_0(x) \leftarrow -x^{n-1}$.
(3-b) For $j = 0$ to $m - 1$, do

i. As in (2-b), set $f_{n_j}(x) = x^{n_j} + 1$ with $n_j = n/2^j$. Compute $A_j(x) = V_j(x) \cdot V_j(-x) \bmod f_{n_j}(x) = \sum_{i=0}^{n_j-1} a_i x^i$, $B_j(x) = U_j(x) \cdot V_j(-x) + U_j(-x) \cdot V_j(x) \bmod f_{n_j}(x) = \sum_{i=0}^{n_j-1} b_i x^i$ and $C_j(x) = W_j(x) \cdot V_j(-x) + W_j(-x) \cdot V_j(x) \bmod f_{n_j}(x) = \sum_{i=0}^{n_j-1} c_i x^i$.
ii. $U_{j+1}(x) \leftarrow \sum_{k=0}^{n_j/2-1} b_{2k} x^k$, $V_{j+1}(x) \leftarrow \sum_{k=0}^{n_j/2-1} a_{2k} x^k$, $W_{j+1}(x) \leftarrow \sum_{k=0}^{n_j/2-1} c_{2k} x^k$.

(3-c) Then $d = V_m, g_1 = U_m, h_1 = W_m$, and hence $w_0 = -g_1/n, w_1 = -h_1/n$.

4: Set $r = w_1/w_0 \bmod d$ and verify that $r^n \equiv -1 \bmod d$. If $r^n \equiv -1 \bmod d$, set $\mathsf{pk} = (d, r, n, s)$. Otherwise, go back to Step 1.
5: For $i = 0$ to $n-1$, find $w_i = r^i \cdot w_0 \bmod d$ with $\gcd([w_i]_d, s) = 1$. If $\gcd([w_i]_d, s) = 1$, set w_i as the secret key sk (in using the condition (♠), we set w_0 as sk).

4 Application to Biometric Authentication

In this section, we propose a privacy-preserving biometric authentication protocol using our variant SHE scheme with our packing method. We also compare our protocol with the related protocols described in §1.1, and demonstrate the efficiency of our packing method.

4.1 Protocol Using SHE Scheme Based on Ideal Lattices

Unlike SCiFI and the protocol of [2], our protocol involves three parties, a client server $\mathcal{C}$, a computation server $\mathcal{S}$, and an authentication server $\mathcal{A}$. We assume that the authentication server $\mathcal{A}$ is a trusted party to manage the secret key of homomorphic encryption. Our protocol is based on the work in [16] using

generic 2-DNF (disjunctive normal form) homomorphic encryption such as the BGN scheme [3], which can support additions and one depth multiplications on encrypted data. In contrast to their work, we use the SHE scheme based on ideal lattices. In the following, we give a construction of our protocol of biometric authentication with ID, namely, one-to-one authentication protocol:

Setup Phase. The authentication server $\mathcal{A}$ generates the public key pk and the secret key sk of our variant scheme. Then the server $\mathcal{A}$ distributes only the public key pk to both the client server $\mathcal{C}$ and the computation server $\mathcal{S}$.

Enrollment Phase

1. The client server $\mathcal{C}$ generates a feature vector A of 2048-bit from client's biometric data such as fingerprints, encrypts A using our packing method of type 1 (see Definition 3), and sends the encrypted feature vector $\mathsf{vEnc}_1(A)$ with client's ID to the computation server $\mathcal{S}$.
2. The computation server $\mathcal{S}$ stores the encrypted feature vector $\mathsf{vEnc}_1(A)$ in the database D as a template with client's ID.

Authentication Phase

1. As in the enrollment phase, the client server $\mathcal{C}$ generates a feature vector B of 2048-bit from client's biometric data, encrypts B using our packing method of type 2 (note that it is different from the encryption method in the enrollment phase), and sends the encrypted feature vector $\mathsf{vEnc}_2(B)$ with client's ID to the computation server $\mathcal{S}$.
2. The computation server $\mathcal{S}$ extracts the template $\mathsf{vEnc}_1(A)$ corresponding to client's ID from the database D. Then the server $\mathcal{S}$ computes the encrypted Hamming distance ct_H of $\mathsf{vEnc}_1(A)$ and $\mathsf{vEnc}_2(B)$ defined by the equation (9), and sends only the encrypted data ct_H to the authentication server $\mathcal{A}$.
3. The authentication server $\mathcal{A}$ decrypts the encrypted data ct_H with the secret key sk to obtain the Hamming distance $d_H(A, B)$. Finally, the server $\mathcal{A}$ returns the authentication result 'OK' (resp. 'NG') if $d_H(A, B) \leq \sigma$ (resp. otherwise), where let σ denote pre-defined threshold.

In our protocol, all feature vectors handled by the computation server $\mathcal{S}$ are protected by homomorphic encryption, and hence we hope that we could use the cloud as the server $\mathcal{S}$ for outsourcing storage of templates and computation resources of secure Hamming distance. Furthermore, since the method $\mathsf{vEnc}_1(A)$ to encrypt a feature vector A in the enrollment phase is asymmetric to the method $\mathsf{vEnc}_2(B)$ in the authentication phase, our protocol is expected to have the additional advantage that identity theft is harder than the conventional protocols even if templates are leaked to attackers (even if an attacker steals a template $\mathsf{vEnc}_1(A)$ from the database $\mathcal{D}$ and sends it to the computation server $\mathcal{S}$ instead of $\mathsf{vEnc}_2(B)$ in the authentication phase, the authentication would fail with very high probability since the decryption result in the authentication server $\mathcal{A}$ is not by the Hamming distance between A and A due to lack of $\mathsf{vEnc}_2(A)$). From the above discussions, we believe that our protocol would be secure as long as the authentication server $\mathcal{A}$ manages the secret key sk correctly.

4.2 Choosing Parameters and Security Analysis

In this section, we describe a method to choose key parameters for secure Hamming distance described in §3.2, and estimate their security level.

Key Parameters. We give key parameters (n, t, s) suitable for secure Hamming distance ct_H given in the equation (9). Our chosen key parameters are

$$(n, t, s) = (2048, 37, 2048) \text{ and } (4096, 37, 2048). \tag{13}$$

These key parameters are estimated to have more than 80-bit security against exhaustive-search and birthday attacks on $v(x)$ generated under the condition (♠) since $2^t = 2^{37}$ is enough large compared with $4n \cdot s = 2^{24}$ or 2^{25} (for example, exhaustive-search attack takes $(2^t/(4n \cdot s))^n = (2^{13})^{2048}$ or $(2^{12})^{4096}$).

The method to choose the key parameters (13) is as follows: Since $0 \leq d_H(A, B) \leq 2048$, we set $s = 2048$ to avoid expensive carry operations. To encrypt a 2048-bit binary vector in a single ciphertext, we should take $n \geq 2048$ and we only consider the two cases $n = 2048$ and 4096 for the practical use (see also Definition 3). Let $\boldsymbol{a}_H$ be the masked plaintext corresponding to the encrypted Hamming distance ct_H as in §3.2. We let $\boldsymbol{a}_1$ and $\boldsymbol{b}_1$ denote the masked plaintexts corresponding to $\mathsf{vEnc}_1(A)$ and $\mathsf{vEnc}_2(B)$, respectively. We also let $\boldsymbol{a}_2$ and $\boldsymbol{b}_2$ denote the masked plaintexts corresponding to the homomorphically operated ciphertexts $C_2 * \mathsf{vEnc}_1(A)$ and $C_1 * \mathsf{vEnc}_2(B)$, respectively. Then we have

$$\boldsymbol{a}_H = \boldsymbol{a}_2 + \boldsymbol{b}_2 - 2\boldsymbol{a}_1 \times \boldsymbol{b}_1 \in R$$

by the equation (9). By evaluating the ∞-norm size of $\boldsymbol{a}_H$, we determine suitable t for decryption success of the ciphertext ct_H. For any two elements $\boldsymbol{a}, \boldsymbol{b} \in R$, we have $||\boldsymbol{a} + \boldsymbol{b}||_\infty \leq ||\boldsymbol{a}||_\infty + ||\boldsymbol{b}||_\infty$, $||\boldsymbol{a} \times \boldsymbol{b}||_\infty \leq n \cdot ||\boldsymbol{a}||_\infty \cdot ||\boldsymbol{b}||_\infty$ (see [21, Lemma 3.2]). Since we only consider $n = 2048$ and 4096, we have

$$\begin{aligned} ||\boldsymbol{a}_H||_\infty &\leq ||\boldsymbol{a}_2||_\infty + ||\boldsymbol{b}_2||_\infty + 2n \cdot ||\boldsymbol{a}_1||_\infty \cdot ||\boldsymbol{b}_1||_\infty \\ &\leq 2n \cdot (2^{11} + 1) + 2n \cdot (2^{11} + 1)^2 \leq 2^{26} + 2^{35}. \end{aligned}$$

Note that we have $||\boldsymbol{a}_1||_\infty, ||\boldsymbol{b}_1||_\infty \leq 2^{11} + 1$ and $||\boldsymbol{a}_2||_\infty, ||\boldsymbol{b}_2||_\infty \leq n \cdot (2^{11} + 1)$ due to $s = 2^{11}$. By the ∞-norm evaluation of Proposition 1, we estimate that the decryption of ct_H succeeds if we take $t \geq 37$, and then we fix $t = 37$ for better performance and shorter ciphertexts.

Security Analysis against Lattice Reduction Attack. By using lattice reduction algorithms with considerably small root Hermite factor (see §2.1 for the root Hermite factor), we can recover the plaintext from a ciphertext without the secret key, which we simply call a lattice reduction attack. We note that the hardness of the lattice reduction attack is closely related to the ICP assumption, on which the provable security of our variant scheme relies (see Remark 1). We estimate the security level of our key parameters (13) against the attack.

Its security is based on the hardness of the lattice problem γ-BDDP [14, Section 2.2] with the parameter

$$\gamma = \frac{d^{1/n}}{\min ||\boldsymbol{a}||} \approx \frac{(2^{tn})^{1/n}}{s \cdot \sqrt{2n/3}} = \frac{2^t}{s \cdot \sqrt{2n/3}}, \tag{14}$$

where $\min ||\boldsymbol{a}||$ denotes the minimal Euclidean norm of the masked plaintexts. Note that $\min ||\boldsymbol{a}|| = ||\boldsymbol{a}_1||$ or $||\boldsymbol{b}_1|| \approx s \cdot \sqrt{2n/3}$ due to the probability $q = \frac{1}{3}$, where $\boldsymbol{a}_1$ (resp. $\boldsymbol{b}_1$) denotes the masked plaintext corresponding to $\mathsf{vEnc}_1(A)$ (resp. $\mathsf{vEnc}_2(B)$). BDDP is intuitively the analogue of the lattice problem unique-SVP, and it can be solved by lattice reduction algorithms with the root Hermite factor smaller than γ [12].

The case $(n, t, s) = (2048, 37, 2048)$. We have $\gamma = 1.0071^n$ by the equation (14). Since the root Hermite factor of LLL (resp. BKZ with $\beta = 20$) is practically 1.02^n (resp. 1.0128^n) on average (see experimental results of the work in [12]), it cannot be solved by LLL and BKZ with $\beta = 20$. At present, BKZ 2.0 is the state-of-the-art implementation of BKZ with large $\beta \geq 50$. The approximate root Hermite factor achieved by BKZ 2.0 is shown in [8, Table 2]. According to [8, Table 2], the BKZ blocksize $\beta \approx 160$ is needed to solve this case (the root Hermite factor of BKZ 2.0 with $\beta = 168$ is predicted to be 1.007^n). The approximate running time of BKZ 2.0 is given by $z \times \dim \times e$, where z is the number of rounds, "dim" is the lattice dimension and e is the approximate running time of the enumeration subroutine of blocksize β (see [8] for details). Since a few rounds are sufficient to achieve an enough performance of BKZ 2.0, we assume $z = 1$ for simplicity. As for e, Chen and Nguyen in [8, Table 3] gave an estimated upper bound on the cost of the enumeration subroutine. From [8, Table 3], we have that the enumeration subroutine of BKZ 2.0 with $\beta = 160$ is estimated at $2^{84.7}$ nodes at maximum. Since they also reported that one node requires about 200 clock cycles, we estimate that $e \leq 200 \times 2^{84.7} \approx 2^{92.3}$ clock cycles for BKZ 2.0 with $\beta = 160$. Therefore we estimate that the running time of BKZ 2.0 with $\beta = 160$ is at most $2^{11} \times 2^{92.3} = 2^{103.3}$ cycles in dimension 2048, which shows that this case may reach 80-bit security against BKZ 2.0. We remark that this estimation is just an upper bound cost of BKZ 2.0.

The case $(n, t, s) = (4096, 37, 2048)$. We have $\gamma = 1.0034^n$ in this case. The BKZ blocksize $\beta \geq 280$ is needed to solve this case from [8, Table 2] (the root Hermite factor of BKZ 2.0 with $\beta = 286$ is predicted to be 1.005^n). In contrast to [8, Table 3], they in [8, Table 4] also gave an estimated lower bound of the cost of e. According to [8, Table 4], the cost of e for BKZ 2.0 with $\beta = 280$ is estimated at $2^{79.8}$ nodes at lowest. From a similar argument as above, we estimate $e \geq 200 \times 2^{79.8} \approx 2^{87.4}$ clock cycles. Therefore we estimate that the running time of BKZ 2.0 with $\beta = 280$ is at least $2^{12} \times 2^{87.4} \approx 2^{99.4}$ clock cycles in dimension 4096. Hence we conclude that this case is estimated at 80-bit security with an enough margin (we can estimate from [8, Table 3] that this case has more than 192-bit security against BKZ 2.0).

Table 1. Performance and ciphertext size of our variant scheme

Parameters (n, t, s)	Key gen. (Alg. 1)	Precomp. Table	Packed Enc.	Secure Hamming	Dec.	Cipher. size	BDDP instance
$(2048, 37, 2048)$	220 ms	5.20 sec	6.20 ms	4.98 ms	2.51 ms	9.5 KB	1.0071^n
$(4096, 37, 2048)$	870 ms	38.15 sec	19.89 ms	18.10 ms	9.08 ms	19 KB	1.0034^n

4.3 Implementation Results

We implemented our variant scheme for the key parameters (13). The performance and the ciphertext size are shown in Table 1. We remark that the packed encryption vEnc_i is computed by the equation (6) or (7), the secure Hamming distance is computed by the equation (9) using the precomputed integer C_1, and the decryption of the encrypted Hamming distance ct_H is computed by (10) with the secret key sk. In particular, the homomorphic addition takes about 0.004 ms (resp. 0.007 ms), and the homomorphic multiplication takes about 2.46 ms (resp. 9.00 ms) for $(n, t, s) = (2048, 37, 2048)$ (resp. $(n, t, s) = (4096, 37, 2048)$). In our implementation, we used our software library written with assembly language $\mathsf{x86\_64}$ for all computations in the base ring $R/(v(x)) \simeq \mathbb{Z}/d\mathbb{Z}$ (cf. see also [21] for its implementation result of the SHE scheme based on the ring-LWE problem). Our experiments ran on an Intel Xeon X3480 at 3.07 GHz with 16 GByte memory. We also implemented Karatsuba Multiplication and Montgomery Reduction algorithm for efficient multiplication in the base ring $\mathbb{Z}/d\mathbb{Z}$. Furthermore, we used a precomputed table $T = \{1, [r]_d, [r^2]_d, \ldots, [r^{n-1}]_d\}$ to make the encryption and the homomorphic operations much faster.

4.4 Comparison to Related Work

In Table 2, we give a comparison of our protocol using a lattice of 4096 dimension with the related protocols described in §1.1, with respect to the performance and the ciphertext size, which are main issues for the practical use of homomorphic encryption as remarked in §1 (all encryption schemes in Table 2 are estimated to have 80-bit security). From Table 2, our protocol has faster performance and shorter size than SCiFI and the protocol of [2]. In particular, our protocol is about 8 times faster and about 14 times shorter than the protocol of [2] when we ignore the difference of the PC performance. This is mainly due to our packing method since a feature vector can only be encrypted bit by bit due to the property of additive encryption schemes in SCiFI and the protocol of [2].

Furthermore, let us compare our protocol with one using the BGN scheme [3] such as the work in [16]. The BGN scheme is based on bilinear pairings on elliptic curves, and it requires a pairing computation for its homomorphic multiplication. According to [21, Section 1.2], the homomorphic multiplication of the BGN scheme is very slower than that of lattice-based homomorphic encryption. Furthermore, we can not use our packing method in the BGN scheme, and hence it needs 2048 homomorphic multiplications for secure Hamming distance of two vectors of 2048-bit. Even if we use very fast implementation taking 1 ms for

Table 2. Comparison to protocols using homomorphic encryption

Protocols (feature vector size)	Performance of Secure Hamming	Size increase rate by encryption* (cipher. size)	Homomorphic encryption scheme
SCiFI [23] (900-bit)	310 ms$^{(a)}$	2048 times (230 KByte)	Paillier-1024 (additive scheme)
Protocol of [2] (2048-bit)	150 ms$^{(b)}$	1024 times (262 KByte)	DGK-1024 (additive scheme)
Ours (2048-bit)	**18.10 ms**$^{(c)}$	**$\approx$ 80 times (19 KByte)**	**ideal lattices-4096 (SHE scheme)**

*denotes the ratio of (encrypted feature vector size)/(plain feature vector size)
$^{(a)}$on an 8 core machine of 2.6 GHz AMD Opteron processors with 1 GByte memory
$^{(b)}$on an Intel Core 2 Duo 2.13 GHz with 3 GByte memory
$^{(c)}$on an Intel Xeon X3480 at 3.07 GHz with 16 GByte memory

one pairing computation, it takes about 2048 ms $\approx$ 2 sec for secure Hamming distance. Therefore we estimate that our protocol is at least about 100 times faster than one using the BGN scheme (our protocol is also estimated to have shorter size than one using the BGN scheme due to our packing method).

References

1. Belguechi, R., Alimi, V., Cherrier, E., Lacharme, P., Rosenberger, C.: An overview on privacy preserving biometrics, `http://cdn.intechopen.com/pdfs/17038/InTech-An_overview_on_privacy_preserving_biometrics.pdf`
2. Blanton, M., Gasti, P.: Secure and efficient protocols for iris and fingerprint identification. In: Atluri, V., Diaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 190–209. Springer, Heidelberg (2011)
3. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
4. Brakerski, Z., Gentry, C., Halevi, S.: Packed ciphertexts in LWE-based homomorphic encryption. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 1–13. Springer, Heidelberg (2013)
5. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: Innovations in Theoretical Computer Science–ITCS 2012, pp. 309–325 (2012)
6. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent message. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011)
7. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Foundations of Computer Science–FOCS 2011, pp. 97–106 (2011)
8. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 1–20. Springer, Heidelberg (2011)
9. Coron, J.-S., Mandal, A., Naccache, D., Tibouchi, M.: Fully homomorphic encryption over the integers with shorter public keys. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 487–504. Springer, Heidelberg (2011)
10. Damgård, I., Geisler, M., Krøigård, M.: Homomorphic encryption and secure comparison. Journal of Applied Cryptography 1(1), 22–31(2008)

11. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010)
12. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 31–51. Springer, Heidelberg (2008)
13. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Symposium on Theory of Computing–STOC 2009, pp. 169–178. ACM (2009)
14. Gentry, C., Halevi, S.: Implementing Gentry's fully-homomorphic encryption scheme. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 129–148. Springer, Heidelberg (2011)
15. Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the AES circuit. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 850–867. Springer, Heidelberg (2012)
16. Hattori, M., Matsuda, N., Ito, T., Takashima, K., Yoneda, T.: Provably-secure cancelable biometrics using 2-DNF evaluation. Journal of Information Processing 20(2), 496–507 (2012)
17. Jain, A.K., Nandakumar, K., Nagar, A.: Biometric template security (review article). EURASIP Journal on Advances in Signal Processing, 1–17 (2008)
18. Jarrous, A., Pinkas, B.: Secure hamming distance based computation and its applications. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 107–124. Springer, Heidelberg (2009)
19. Juels, A., Sudan, M.: A fuzzy vault scheme. Designs, Codes and Cryptography 38(2), 237–257 (2006)
20. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: ACM CCS 1999, pp. 28–36 (1999)
21. Lauter, K., Naehrig, M., Vaikuntanathan, V.: Can homomorphic encryption be practical? In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, pp. 113–124 (2011)
22. Loftus, J., May, A., Smart, N.P., Vercauteren, F.: On CCA-secure somewhat homomorphic encryption. In: Miri, A., Vaudenay, S. (eds.) SAC 2011. LNCS, vol. 7118, pp. 55–72. Springer, Heidelberg (2012)
23. Osadchy, M., Pinkas, B., Jarrous, A., Moskovich, B.: SCiFI - A system for secure face identification. In: IEEE Symposium on Security and Privacy, pp. 239–254 (2010)
24. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
25. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)
26. Ratha, N., Connelle, J., Bolle, R.: Enhansing security and privacy in biometrics-based authentication system. IBM Systems J. 37(11), 2245–2255 (2001)
27. Schoenmakers, B., Tuyls, P.: Efficient binary conversion for paillier encrypted values. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 522–537. Springer, Heidelberg (2006)
28. Smart, N.P., Vercauteren, F.: Fully homomorphic SIMD operations. Designs, Codes and Cryptography (to appear)
29. U.S. Department of Homeland Security, Privacy impact assessment for the biometric storage system, `http://www.dhs.gov/xlibrary/assets/privacy/privacy_pia_cis_bss.pdf` (March 28, 2007)